

KIP-196: Add metrics to Kafka Connect framework

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [Connector Metrics](#)
 - [Common Task Metrics](#)
 - [Source Task Metrics](#)
 - [Worker Metrics](#)
 - [Worker Rebalance Metrics](#)
 - [Configuration](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *Approved*

Discussion thread: [here](#)

JIRA: [here](#)

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

An important part of deploying Kafka Connect is monitoring the health of the workers in a cluster and the connectors and tasks that have been deployed to the cluster. Although producers and consumers used in Kafka Connect can be monitored, the Kafka Connect framework only has a few metrics capturing the number of connectors and tasks for each worker. To augment these existing metrics, we propose to add metrics to monitor more information about the connectors, tasks, and workers. All metrics reported by each worker are scoped by the activities within that worker.

There are several things that are out of scope for this proposal, though they may be addressed in future KIPs. First, this proposal expressly avoids changes to the Connect API, and therefore does not address how connector implementations can define their own connector-specific metrics. Second, Kafka Connect does not have any existing mechanism to aggregate the metrics reported across multiple workers.

Public Interfaces

All of the following will be added via Kafka's metrics library like most of the metrics in the Kafka brokers and other components. The context of all metrics are limited to the worker where the metrics are being reported, and all metrics are defined as attributes on the specified MBean attribute and are measured within the context of a single worker. All metrics defined below are at the INFO recording level.

Connector Metrics

MBean name: `kafka.connect:type=connector-metrics,connector=[-.\w]+)`

Metric/Attribute Name	Description	Implemented
connector-type	The type of the connector, one of: source, sink	1.0.0
connector-class	The name of the connector class	1.0.0
connector-version	The version of the connector class, as reported by the connector in this worker	1.0.0
status	The current status of the connector in this worker, one of: running, paused, stopped	1.0.0

Common Task Metrics

MBean name: `kafka.connect:type=connector-task-metrics,connector=[-.\w]+),task=[-.\w]+)`

Metric/Attribute Name	Description	Implemented
status	The current status of this task, one of: unassigned, running, paused, failed, destroyed	1.0.0

pause-ratio	The fraction of time this task has spent in the paused state.	1.0.0
running-ratio	The fraction of time this task has spent in the running state.	1.0.0
offset-commit-success-percentage	The average percentage of this task's offset commit attempts that succeeded	1.0.0
offset-commit-failure-percentage	The average percentage of this task's offset commit attempts that failed or had an error	1.0.0
offset-commit-max-time-ms	The maximum time in milliseconds taken by this task to commit offsets	1.0.0
offset-commit-avg-time-ms	The average time in milliseconds taken by this task to commit offsets	1.0.0
offset-commit-99p-time-ms	The 99th percentile time in milliseconds spent by this task to commit offsets to Kafka	
offset-commit-95p-time-ms	The 95th percentile time in milliseconds spent by this task to commit offsets to Kafka	
offset-commit-90p-time-ms	The 90th percentile time in milliseconds spent by this task to commit offsets to Kafka	
offset-commit-75p-time-ms	The 75th percentile time in milliseconds spent by this task to commit offsets to Kafka	
offset-commit-50p-time-ms	The 50th percentile (average) time in milliseconds spent by this task to commit offsets to Kafka	
batch-size-max	The maximum size of the batches processed by the connector	1.0.0
batch-size-avg	The average size of the batches processed by the connector	1.0.0

Source Task Metrics

MBean name: `kafka.connect:type=source-task-metrics,connector=([-.\w]+),task=([\d]+)`

Metric /Attribute Name	Description	Implemented
source-record-poll-rate	The average per-second number of records produced/pollled (before transformation) by this task belonging to the named source connector in this worker.	1.0.0
source-record-poll-total	The number of records produced/pollled (before transformation) by this task belonging to the named source connector in this worker, since the task was last restarted.	1.0.0
source-record-write-rate	The average per-second number of records output from the transformations and written to Kafka for this task belonging to the named source connector in this worker. This is after transformations are applied and excludes any records filtered out by the transformations.	1.0.0
source-record-write-total	The number of records output from the transformations and written to Kafka for this task belonging to the named source connector in this worker, since the task was last restarted.	1.0.0
source-record-active-count	The most recent number of records that have been produced by this task but not yet completely written to Kafka.	1.0.0
source-record-active-count-max	The maximum number of records that have been produced by this task but not yet completely written to Kafka.	1.0.0
source-record-active-count-avg	The average number of records that have been produced by this task but not yet completely written to Kafka.	1.0.0
poll-batch-max-time-ms	The maximum time in milliseconds taken by this task to poll for a batch of source records	1.0.0
poll-batch-avg-time-ms	The average time in milliseconds taken by this task to poll for a batch of source records	1.0.0
poll-batch-99p-time-ms	The 99th percentile time in milliseconds spent by this task to poll for a batch of source records	
poll-batch-95p-time-ms	The 95th percentile time in milliseconds spent by this task to poll for a batch of source records	
poll-batch-90p-time-ms	The 90th percentile time in milliseconds spent by this task to poll for a batch of source records	

poll-batch-75p-time-ms	The 75th percentile time in milliseconds spent by this task to poll for a batch of source records	
poll-batch-50p-time-ms	The 50th percentile (average) time in milliseconds spent by this task to poll for a batch of source records	

Sink Task Metrics

MBean name: `kafka.connect:type=sink-task-metrics,connector=([-.\w]+),task=([\d]+)`

Metric /Attribute Name	Description	Implemented
sink-record-read-rate	The average per-second number of records read from Kafka for this task belonging to the named sink connector in this worker. This is before transformations are applied.	1.0.0
sink-record-read-total	The total number of records produced/polled (before transformation) by this task belonging to the named source connector in this worker, since the task was last restarted.	1.0.0
sink-record-send-rate	The average per-second number of records output from the transformations and sent to this task belonging to the named sink connector in this worker. This is after transformations are applied and excludes any records filtered out by the transformations.	1.0.0
sink-record-send-total	The total number of records output from the transformations and sent to this task belonging to the named source connector in this worker, since the task was last restarted.	1.0.0
sink-record-lag-max	The maximum lag in terms of number of records behind the consumer the offset commits are for any topic partitions.	
sink-record-active-count	The most recent number of records that have been read from Kafka but not yet completely committed/flushed /acknowledged by the sink task.	1.0.0
sink-record-active-count-max	The maximum number of records that have been read from Kafka but not yet completely committed/flushed /acknowledged by the sink task.	1.0.0
sink-record-active-count-avg	The average number of records that have been read from Kafka but not yet completely committed/flushed /acknowledged by the sink task.	1.0.0
partition-count	The number of topic partitions assigned to this task belonging to the named sink connector in this worker.	1.0.0
offset-commit-seq-no	The current sequence number for offset commits	1.0.0
offset-commit-completion-rate	The average per-second number of offset commit completions that were completed successfully	1.0.0
offset-commit-completion-total	The total number of offset commit completions that were completed successfully	1.0.0
offset-commit-skip-rate	The average per-second number of offset commit completions that were received too late and skipped/ignored	1.0.0
offset-commit-skip-total	The total number of offset commit completions that were received too late and skipped/ignored	1.0.0
put-batch-max-time-ms	The maximum time in milliseconds taken by this task to put a batch of sinks records	1.0.0
put-batch-avg-time-ms	The average time in milliseconds taken by this task to put a batch of sinks records	1.0.0
put-batch-99p-time-ms	The 99th percentile time in milliseconds spent by this task to put a batch of sinks records	
put-batch-95p-time-ms	The 95th percentile time in milliseconds spent by this task to put a batch of sinks records	

put-batch-90p-time-ms	The 90th percentile time in milliseconds spent by this task to put a batch of sinks records	
put-batch-75p-time-ms	The 75th percentile time in milliseconds spent by this task to put a batch of sinks records	
put-batch-50p-time-ms	The 50th percentile (average) time in milliseconds spent by this task to put a batch of sinks records	
flush-max-time-ms	The maximum time in milliseconds taken by this sink task to pre-commit/flush	
flush-99p-time-ms	The 99th percentile time in milliseconds spent by this sink task to pre-commit/flush	
flush-95p-time-ms	The 95th percentile time in milliseconds spent by this sink task to pre-commit/flush	
flush-90p-time-ms	The 90th percentile time in milliseconds spent by this sink task to pre-commit/flush	
flush-75p-time-ms	The 75th percentile time in milliseconds spent by this sink task to pre-commit/flush	
flush-50p-time-ms	The 50th percentile (average) time in milliseconds spent by this sink task to pre-commit/flush	

MBean name: `kafka.connect:type=sink-task-metrics,connector=([-.\w]+),task=([\d]+),topic=([-.\w]+),partition=([\d]+)`

Metric/Attribute Name	Description	Implemented
sink-record-lag	The latest lag in terms of number of records behind the consumer the offset commits are for the topic partition.	
sink-record-lag-avg	The average lag in terms of number of records behind the consumer the offset commits are for the topic partition.	
sink-record-lag-max	The maximum lag in terms of number of records behind the consumer the offset commits are for the topic partition.	

Worker Metrics

MBean name: `kafka.connect:type=connect-worker-metrics`

Metric/Attribute Name	Description	Implemented
task-count	The number of tasks run in this worker	1.0.0
connector-count	The number of connectors run in this worker	1.0.0
connector-startup-attempts-total	The total number of connector startups that this worker has attempted.	1.0.0
connector-startup-success-total	The total number of connector starts that succeeded.	1.0.0
connector-startup-success-percentage	The average percentage of this worker's connectors starts that succeeded.	1.0.0
connector-startup-failure-total	The total number of connector starts that failed.	1.0.0
connector-startup-failure-percentage	The average percentage of this worker's connectors starts that failed.	1.0.0
task-startup-attempts-total	The total number of task startups that this worker has attempted.	1.0.0
task-startup-success-total	The total number of task starts that succeeded.	1.0.0
task-startup-success-percentage	The average percentage of this worker's task starts that succeeded.	1.0.0
task-startup-failure-total	The total number of task starts that failed.	1.0.0
task-startup-failure-percentage	The average percentage of this worker's task starts that failed.	1.0.0
rest-request-rate	The average per second number of requests handled by the REST endpoints in this worker	
rest-request-total	The total number of requests handled by the REST endpoints in this worker	

Worker Rebalance Metrics

MBean name: `kafka.connect:type=connect-worker-rebalance-metrics`

Metric/Attribute Name	Description	Implemented
leader-name	The name of the group leader	1.0.0
epoch	The epoch or generation number of this worker	1.0.0
completed-rebalances-total	The total number of rebalances completed by this worker.	1.0.0
rebalancing	Whether this worker is currently rebalancing.	1.0.0
rebalance-max-time-ms	The maximum time in milliseconds spent by this worker to rebalance.	1.0.0
rebalance-avg-time-ms	The average time in milliseconds spent by this worker to rebalance.	1.0.0
rebalance-99p-time-ms	The 99th percentile time in milliseconds spent by this worker to rebalance during the last window (defaults to an hour)	
rebalance-95p-time-ms	The 95th percentile time in milliseconds spent by this worker to rebalance during the last window (defaults to an hour)	
rebalance-90p-time-ms	The 90th percentile time in milliseconds spent by this worker to rebalance during the last window (defaults to an hour)	
rebalance-75p-time-ms	The 75th percentile time in milliseconds spent by this worker to rebalance during the last window (defaults to an hour)	
rebalance-50p-time-ms	The 50th percentile (average) time in milliseconds spent by this worker to rebalance during the last window (defaults to an hour)	
time-since-last-rebalance-ms	The time in milliseconds since the most recent rebalance in this worker	1.0.0

Configuration

The distributed and standalone worker configuration files will support the following properties. These exactly match the producer and consumer configurations of the same name. (The first three are already in the distributed worker configuration.)

Configuration Field	Type	Default	Importance	Description
metrics.sample.window.ms	long	30000	low	The window of time in milliseconds a metrics sample is computed over. Must be a non-negative number.
metrics.num.samples	int	2	low	The number of samples maintained to compute metrics. Must be a positive number.
metric.reporters	string	""	low	A list of classes to use as metrics reporters. Implementing the <code>MetricReporter</code> interface allows plugging in classes that will be notified of new metric creation. The <code>JmxReporter</code> is always included to register JMX statistics.
metrics.recording.level	string	"INFO"	low	The highest recording level for metrics. Must be either "INFO" or "DEBUG".

Proposed Changes

We will add the relevant metrics and worker configuration properties as specified in the Public Interfaces section.

Compatibility, Deprecation, and Migration Plan

Existing Connect coordinator metrics will not be changed.

The `metrics.sample.window.ms`, `metrics.num.samples`, and `metric.reporters` configurations already exist in the distributed worker; these will also be added to the standalone worker. The `metrics.recording.level` configuration will be added to both the distributed and standalone worker configurations. All four of these metrics have sensible default values and therefore users do ***not*** need to add or override them in their existing configuration files.

Rejected Alternatives

None